

PC-8023A-C DOT MATRIX

PRINTER USER'S MANUAL

FOR APPLE USERS

Left Margin set uses unenlarged characters

PREFACE

This is a supplement to the PC-8023A-C Dot Matrix Printer User's Manual (hereafter referred to as "the manual") for Apple[®] users. This supplement provides the following:

- A section on DIP switch settings for use with an Apple.
- A section for using the PC-8023A-C (hereafter referred to as "the printer") with the APPLE II[®] plus computer.
- An Applesoft printer test program.

This supplement assumes that the Apple is controlling the printer in JA7 mode. This mode simply means that DIP switches SW1-1, SW1-2, and SW1-3, are OFF and SW2-6 is ON. In JA7 mode the printer only recognizes the seven least significant bits. Although the printer can recognize all eight bits if switch SW2-6 is OFF, not all parallel interfaces for the Apple can output eight data bits. Even though this manual shows how to control the printer with seven bits instead of eight, the user does not sacrifice any of the capabilities of the printer despite using one less bit.

A brief glossary of terms used in this manual is provided in the back. To those new to computers and printers and not familiar with associated terminology, it might be helpful to look at this glossary of terms before proceeding any further.

TABLE OF CONTENTS

	PAGE
1) DIP Switch Settings For Using the PC-8023A-C With an Apple II plus	1
1.1) DIP-switch Settings and their Functions	1
1.2) DIP Switch Summary for the Apple II plus	5
2) Control of the PC-8023A-C Using Applesoft BASIC	6
2.1) Character Set Selection	6
2.1.1) Selecting the ASCII Character Set	6
2.1.2) Selecting the CG Graphics Mode ..	7
2.1.3) Selecting the Greek Mode	7
2.1.4) Selecting the Katakana Mode	8
2.2) Selecting the Dot-Image Graphics Mode	11
2.3) Changing the Font Type/ Character Size	14
2.3.1) Pica Characters (10 cpi)	15
2.3.2) Elite Characters (12 cpi)	15
2.3.3) Condensed Characters (17 cpi) .	15
2.3.4) Proportional Characters	16
2.4) Printing Enlarged Characters	16
2.5) Character Enhancement	18
2.6) Inter-character Spacing for Proportionally Spaced Printing (Dot Spacing)	19
2.7) Underline Printing	19
2.8) Setting the Left Margin	21
2.9) Horizontal Tabulation	21
2.10) Line Spacing for N Lines	24

2.11)	Vertical Tabulation	24
2.12)	Programmable Line Spacing	29
2.13)	Changing the Line Feed Direction .	30
2.14)	Incremental Mode/Logical- Seeking Mode Selection	30
2.15)	Back Spacing	31
2.16)	Printer Select and Deselect Mode .	31
2.17)	The Cancel Data Command	33
2.18)	The Line Feed Command	33
2.19)	The Form Feed Command	33
3)	An Applesoft Printer Test Program ..	35
4)	Sample Output Demonstrating Some of the PC-8023A-Cs Capabilities	37
5)	Summary of ESCape Codes and Function Codes	39
6)	Glossary of Terms	44
7)	The NEC PC-8023A-C Character Set (with ASCII table and foreign characters)	45

1) DIP-SWITCH SETTINGS FOR USING THE
PC-8023A-C WITH AN APPLE II PLUS

Section 1 is divided into two subsections. The first (1.1) is lengthy and explains all the DIP-switch settings for the printer, while the second (1.2) is a quick summary of typical DIP-switch settings for using the printer with the Apple. If the user is anxious to get the printer up and working, then he/she can skip section 1.1 and just follow section 1.2.

1.1) DIP-SWITCH SETTINGS AND THEIR FUNCTIONS

CONVENTIONS USED BELOW:

- st: Factory-set to these positions
- apple: Suggested switch setting for Apple users
- OFF: This means the switch should be in the open position.
- ON: This means the switch should not be in the open position.

=====

A) SELECTING COUNTRY

<u>COUNTRY</u>	<u>SW1-1</u>	<u>SW1-2</u>	<u>SW1-3</u>	
JA	OFF	OFF	OFF	apple
US	OFF	ON	OFF	st
UK	ON	ON	OFF	
GE	OFF	OFF	ON	
SW	ON	OFF	ON	

* Combinations not listed above are equivalent to the SW specification. (See section 4 for details on foreign characters.)

=====

B) SELECTING THE NUMBER OF LINES
BETWEEN TOP OF FORMS (TOFs)

NUMBER OF LINES

BETWEEN TOFs SW1-4

66 OFF st, apple

72 ON

* 66 lines/page is for 11 inch paper
(vertically).

=====

C) PROCESSING DC1 AND DC3

PROCESSING SW1-5

Invalid ON

Valid OFF st, apple

* When this switch is in the OFF position the
printer recognizes the ASCII control
characters DC1 and DC3 for selecting and
deselecting the printer. The function of
this switch setting is further explained in
section 2.16.

=====

D) SELECTING THE LINE FEED AT FULL-
CHARACTERS PRINT IN ONE LINE

FUNCTION SW1-6

With LF ON st, apple

Without LF OFF

=====

E) SELECTING THE PRINT COMMAND CODE
(VALID IN LOGICAL-SEEKING MODE ONLY)

COMMAND CODE SW1-7

CR only OFF st, apple

CR, LF, VT ON
and FF

=====

F) SELECTING THE CR FUNCTION

<u>FUNCTION</u>	<u>SW1-8</u>
CR	OFF st, apple
CR+LF	ON

=====

G) SELECTING THE ZERO INDICATION

<u>INDICATION</u>	<u>SW2-1</u>
0	OFF
Ø	ON st, apple

* This setting is purely a matter of personal preference. If you prefer that zeros be printed with a slash through them, then the switch should be ON.

=====

H) SELECTING THE DEVICE ADDRESS

<u>CONDITION</u>	<u>SW2-2</u>
Valid	ON
Invalid	OFF st, apple

* Normally this should be kept OFF. See section "I" below and section 2.16 for more information on the function of this switch setting.

=====

I) SETTING THE DEVICE ADDRESS

<u>DEVICE NUMBER</u>	<u>SW2-3</u>	<u>SW2-4</u>
0	OFF	OFF st, apple
1	ON	OFF
2	OFF	ON
3	ON	ON

* If SW2-2 is OFF these switch settings are unimportant. These switches give the printer an address so that it can be selected or deselected via escape sequences if the device select switch (SW2-2) is ON. See section 2.16 for more information on the function of these switches.

=====

J) SETTING THE PRINT MODE AT POWER-ON TIME

<u>MODE</u>	<u>SW2-5</u>	
Proportional	ON	
Pica	OFF	st, apple

=====

K) SETTING 7- OR 8-BIT DATA

<u>DATA</u>	<u>SW2-6</u>	
8 bits	OFF	st
7 bits	ON	apple

- * Even though the printer can be set to recognize all eight bits sent to it, not all parallel interfaces for the Apple send the most significant bit as data. Also because Applesoft outputs the most significant bit high it is not easy to use the printer directly from Applesoft with an 8-bit interface and SW2-6 OFF. This supplement assumes this switch is ON (and switches SW1-1,2,3 are OFF) and it is recommended that these suggested switch settings be followed.

=====

L) SETTING PRINTER SELECT OR DESELECT AT POWER-ON TIME.

<u>MODE</u>	<u>SW2-7</u>	
Select	ON	apple
Deselect	OFF	st

- * If this is not in the ON position the printer will be off-line when the printer is turned on. This means that if SW2-7 is OFF then after the printer is turned on the user has to select the printer by manually pressing the select button on the printers front pannel (or sending an ASCII DC1 code to the printer, which will select the printer if SW1-5 is OFF).

=====

M) RESERVED (SW2-8)

=====

1.2) DIP-SWITCH SUMMARY FOR USE WITH THE APPLE II PLUS

Typical DIP-switch settings for the PC-8023A-C when used with the Apple are:

SW1- 6: ON

SW2- 1,6,7: ON

ALL OTHER SWITCHES OFF (OPEN)

2) CONTROL OF THE PC-8023A-C USING APPLESOFT BASIC

This is an alternate section for Apple users to be substituted for the section in the manual entitled "CONTROL OF THE PC-8023A-C USING NBASIC".

2.1) CHARACTER SET SELECTION

Part of the PC-8023A-C's versatility is the selection of character sets the user can choose from. Section 2.1.x explains how to select the different sets listed in the table below.

<u>PRINTABLE CHARACTERS</u>	<u>RANGE (Hex)</u>	
ASCII	20-7E	Only one
Character Generator (CG) Graphics	20-5F	character
Greek Mode	20-5F	set can be
Katakana ++	20-5F	selected at
		a time in
		JA7 mode.

++ Katakana can only be selected if switches SW1-1 to 3 are all OFF. One SIDE EFFECT of this switch setting is that the back slash character becomes the Yen sign "¥".

2.1.1) SELECTING THE ASCII CHARACTER SET

The printable ASCII character set ranges from 32 to 126 decimal (20-7E hex). These characters are printed below:

```
!"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNPOQRSTUVWXYZ[\]^_
`abcdefghijklmnopqrstuvwxyz{|}~
```

To select the ASCII character set (also referred to as the "alphanumeric/symbolic mode" by the manual) the following escape sequence should be used:

ASCII: ESC \$

DECIMAL: 27 36

HEX: 1B 24

==> FROM APPLESOFT: PRINT CHR\$(27);"\$";

The ASCII SI code (decimal 15, hex 0F) may also be used in JA7 mode.

The ASCII character set is automatically selected when power is turned on. (This code can be used to reselect the ASCII character set if it has been deselected by another control code.)

2.1.2) SELECTING THE CG GRAPHICS MODE

The CG (Character Generator) graphics set is a set of printable symbols that can be used separately, or combined to create crude graphics. In JA7 mode these characters will overlap ASCII and range from 32 to 95 decimal (20-5F hex) after they are selected (as described below).

The CG graphics characters are printed below:

(The last eight of these characters are blank.)

To select the CG graphics mode when in JA7 mode the following escape sequence should be used:

ASCII: ESC #

DECIMAL: 27 35

HEX: 1B 23

```
==> FROM APPLESOFT:  PRINT CHR$(27);"#":
```

To reselect ASCII (i.e. to print text) use "ESC \$" as described above in section 2.1.1.

2.1.3) SELECTING THE GREEK MODE

The Greek character set is a set of characters including the lower case Greek alphabet, several uppercase Greek characters, superscripts and several mathematical symbols. In JA7 mode these characters will overlap ASCII and range from 32 to 95 decimal (20-5F hex) after they are selected (as described below). The characters ranging from 96 to 127 decimal (60-7F hex) become the last 32 CG graphics characters when printed in the Greek mode.

The Greek character set is printed below:

2.*3/0+KJ←→+)(%003Y456€ρσψΩΓΟδκΣ
 αΥΔβ€ηθ1±υπΛ2⊕-тΦωξ√789ιφ≠χ°0ζλμ

To select the Greek mode when in JA7 mode the following escape sequence should be used:

ASCII: ESC &

DECIMAL: 27 38

HEX: 1B 26

==> FROM APPLESOFT: PRINT CHR\$(27);"&;

To reselect ASCII (i.e. to print text) use "ESC \$" as described above in section 2.1.1.

EXAMPLE 2.1.3

```

1 REM EXAMPLE 2.1.3 BY JAY ZIPNICK
2 REM PRINTER INTERFACE IN SLOT 1
10 PR#1
20 REM
30 REM PRINT THE GREEK ALPHABET
40 REM
50 PRINT CHR$(27);"&;":REM SELECT GREEK
60 FOR I=1 TO 24
70 READ L: PRINT CHR$(L);" ";
80 NEXT I
90 REM
100 REM RESELECT ASCII AND FORCE OUT BUFFER
110 REM
120 PRINT CHR$(27);"$";
130 PR#0
140 END
150 DATA 64,67,50,61,54,93,69,70
160 DATA 87,62,94,95,65,68,60,74
170 DATA 55,56,79,73,88,90,57,81

```

The output of the above program is printed below:

α β γ δ ε ζ η θ ι κ λ μ ν ξ ο π ρ σ τ υ φ χ ψ ω

NOTE

Notice that the above program puts the printer in the Greek mode before printing the Greek alphabet and restores ASCII when it is done. Although it is not necessary to restore the printer back to its default settings, it is a good practice to do so. If for example the printer were not restored to ASCII after the alphabet was printed, listing the program on the printer would "list the program in Greek".

2.1.4) SELECTING THE KATAKANA MODE

The Katakana (Japanese) character set overlaps ASCII in JA7 mode (when selected as described below) and ranges from 32 to 95

decimal (20-5F hex). When selected in JA7 mode, the 32 characters 96 to 127 decimal, (60-FF hex) become the last 32 CG graphics characters.

The Katakana characters are printed below:

。「」、・ヲアイウエオヤヨヅーアイウエオカキクケコサシスセソ
タチツテトナニヌネノハヒフヘホマミムメモヤ1ヨラリルレロワン”。

In JA7 mode the Katakana character set is selectable by the control code given below:

ASCII:	^N	(S0)	<=====
			< this is
DECIMAL:	14		< for JA7
			< mode only.
HEX:	0E		<=====

==> FROM APPLESOFT: PRINT CHR\$(14);

If the printer is not in JA7 mode (SW1-1,2,3: OFF, SW2-6: ON) this code will cause character enlargement. If it is in JA7 mode, ASCII can be reselected via "ESC \$" as described in section 2.1.1.

EXAMPLE 2.1.1-4

```

1 REM EXAMPLE 2.1.X BY JAY ZIPNICK
2 REM PRINTER INTERFACE IN SLOT 1
10 PR#1
20 REM
30 REM PRINT TABLE OF JA7 CHARACTERS
40 REM
50 PRINT "COLUMNS = ASCII, CG, GREEK, KATAKANA"
60 E$ = CHR$(27): REM ESCAPE CHAR
70 PRINT E$;"X": REM UNDERLINE
80 FOR HEADING = 2 TO 7
90 PRINT SPC(11);HEADING;
100 NEXT
110 PRINT " ";E$;"Y": REM TURN OFF UNDERLINING
120 REM
130 REM
140 FOR LINE = 0 TO 15
150 LINE$ = RIGHT(" "+STR$(LINE),4)
160 PRINT LINE$;SPC(4);
170 FOR COLUMN = 2 TO 7
180 CH$ = CHR$(LINE + 16 * COLUMN)
190 REM ASCII, CG, GREEK, KATAKANA
200 PRINT CH$;" ";
210 PRINT E$;"#";CH$;
220 PRINT E$;"& ";CH$;" ";
230 PRINT CHR$(14);CH$;E$;"$";SPC(5);
240 NEXT COLUMN
250 PRINT
260 NEXT LINE
270 PR#0
280 END

```

The output of this program is printed below.

COLUMNS = ASCII, CG, GREEK, KATAKANA

	2	3	4	5	6	7
0	—	⊥ ∞ —	@ = α ρ	P X Φ Ξ	、 、 二 二	p p X X
1	! — ∑ .	1 τ ϑ ρ	A ρ ν ρ	Q — ω ρ	a a ρ ρ	q q — —
2	" — . ρ	2 ρ γ ι	B ρ Δ ρ	R ρ ≈ ρ	b b ρ ρ	r r ρ ρ
3	# — * ρ	3 ρ ρ ρ	C ρ ρ ρ	S ρ ρ ρ	c c ρ ρ	s s ρ ρ
4	\$ — ∑ \	4 — 5 I	D ρ ρ ρ	T — 7 ρ	d d ρ ρ	t t ρ ρ
5	% — / .	5 — 6 ρ	E ρ ρ ρ	U ρ ρ ρ	e e ρ ρ	u u ρ ρ
6	& — . ρ	6 ρ ρ ρ	F ρ ρ ρ	V ρ ρ ρ	f f ρ ρ	v v ρ ρ
7	/ — ρ ρ	7 ρ ρ ρ	G ρ ρ ρ	W ρ ρ ρ	g g ρ ρ	w w ρ ρ
8	(ρ ρ ρ	8 ρ ρ ρ	H ρ ρ ρ	X ρ ρ ρ	h h ρ ρ	x x ρ ρ
9) ρ ρ ρ	9 ρ ρ ρ	I ρ ρ ρ	Y ρ ρ ρ	i i ρ ρ	y y ρ ρ
10	* ρ ρ ρ	: ρ ρ ρ	J ρ ρ ρ	Z ρ ρ ρ	j j ρ ρ	z z ρ ρ
11	+ ρ ρ ρ	; ρ ρ ρ	K ρ ρ ρ	[ρ ρ ρ	k k ρ ρ	((ρ ρ
12	, ρ ρ ρ	< ρ ρ ρ	L ρ ρ ρ	¥ ρ ρ ρ	l l ρ ρ	ρ ρ
13	— ρ ρ ρ	= ρ ρ ρ	M ρ ρ ρ] ρ ρ ρ	m m ρ ρ)) ρ ρ
14	. ρ ρ ρ	> ρ ρ ρ	N ρ ρ ρ	^ ρ ρ ρ	n n ρ ρ	~ ~ ρ ρ
15	/ + % ρ	? ρ ρ ρ	O \ ρ ρ	_ ρ ρ ρ	o o \ \	

2.2) SELECTING THE DOT-IMAGE GRAPHICS MODE

Dot-image graphics is a mode in which it is possible to control every dot the PC-8023A-C prints. Every vertical column of dots printed can be represented as a byte of information.

```

LEAST SIGNIFICANT BIT: -> ● ● ● ● 0 0 ● ● 0 ● 0 0
                          ● 0 ● 0 0 0 ● 0 ● 0 0 0
                          ● ● 0 0 ● 0 0 ● 0 ● 0 ●
(dark dots are printed  ● 0 0 0 0 0 ● 0 ● ● 0 ●
empty dots are not)    ● 0 ● 0 0 0 ● ● ● 0 0 0
                          ● ● 0 ● 0 0 0 ● 0 ● ● 0
                          ● 0 ● ● 0 0 ● 0 ● 0 ● ●
MOST SIGNIFICANT BIT:  -> 0 0 0 0 0 0 0 0 0 0 0 0

```

HEX: 7F 25 53 61 04 00 5B 35 5A 2D 60 4C

DECIMAL: 127 37 83 97 4 0 91 53 90 45 96 76

In the above example all the bytes have the most significant bit (MSB) off. This is because it is assumed that the printer is being used with a seven bit interface. If the user can send eight bits to the printer he can control the lower dot printed.

To select the dot-image graphics mode the following escape sequence should be used:

ASCII: ESC S n1 n2 n3 n4

DECIMAL: 27 83 d1 d2 d3 d4

HEX: 1B 53 h1 h2 h3 h4

==> FROM APPLESOFT: PRINT CHR\$(27);"S";"n1n2n3n4";

The four characters after the <escape> "S" are FOUR DECIMAL DIGITS EXPRESSED IN ASCII which is the number of bytes to be printed in dot-image graphics.

The maximum number of horizontal dots per line can be determined by the table below:

PRINT MODE	MAXIMUM NUMBER OF	
	HORIZONTAL DOTS PER LINE *	DOTS PER INCH *
PICA	640	80
ELITE	768	96
CONDENSED	1088	136
PROPORTIONAL	1280	160

* If the printer is in enlarged mode then divide these numbers by two. If the printer is in enhanced mode then every dot printed

is doubled (a second dot is printed partially overlapping the first dot).

NOTE

When in dot-image graphics mode the printer will print dots over all of the vertical positions in which a bit is high, THIS INCLUDES THE MOST SIGNIFICANT BIT. This is REGARDLESS of whether 7 or 8 data bits are selected by switch SW2-6!!!

EXAMPLE 2.2

Using dot-image graphics, print the "smile" shown enlarged below.

1 2 3 4 5 6 7 8 9 10			
	■ ■		(1)
	■ ■	■ ■	(2)
			(4)
			(8)
	■ ■		(16)
	■ ■	■ ■	(32)
		■ ■ ■ ■	(64)
1 2 3 4 5 6 7 8 9 10			

(In this "smile" every block represents an enlarged dot-image graphics dot)

To do this it is first required to send <ESC> "S", followed by "0010" to the printer. This will tell the printer to enter dot-image graphics mode for the next ten bytes. The second step is to determine the bytes that represent each column of dots and send them to the printer. The decimal powers of two are provided on the right of the smile to make it easier to do this. By adding up the powers of two where a dot should be printed (where a binary "1" exists) we get: 16, 48, 35, 67, 64, 64, 67, 35, 48, and 16. (If the 64s are changed to 72s then the smile will have a nose.)

Below is an Applesoft program to print the smile in dot-image graphics.

```

1 REM EXAMPLE 2.2  BY JAY ZIPNICK
2 REM PRINTER INTERFACE IN SLOT 1
10 PR#1
20 REM SELECT DOT-IMAGE GRAPHICS
30 PRINT CHR$(27);CHR$(83);"0010";
40 REM
50 REM  PRINT A SMILE
60 REM
70 FOR I=1 TO 10
80 READ BYTE: PRINT CHR$(BYTE);
90 NEXT I
100 REM
110 REM  OUTPUT <CR> TO PRINT OUT BUFFER
120 REM
130 PRINT
140 PR#0
150 END
160 DATA 16,48,35,67,64,64,67,35,48,16

```

The output of this program is printed below:

☺

Below is another example of dot-image graphics output.

```

oooooooooooooooooooo
☺ ☺ ☺ ☺ ☺ ☺

```

FURTHER NOTES ON DOT-IMAGE GRAPHICS

To use dot-image graphics without gaps between lines, the printers programmable line spacing feature should be used (see section 2.12). This way the amount of space between each printed line can be controlled or eliminated.

Another detail to consider is restrictions on the number of characters per line allowed by the interface card (if there are any restrictions at all). SOME interface cards force a carriage return after a certain number of characters are sent to the printer (this is often programmable by the user through POKE commands or command characters). The interface card does not know that graphics are being printed on the printer, it just keeps count of the number of characters sent through since the last <CR>. If a <CR> is forced out after a certain number of characters are outputted, then an extra byte, 0D hex, 13 decimal (or two extra bytes if a line feed is also forced out) is outputted and printed as graphics, messing up what you intended to print. If your interface card forces a <CR> then it is important to correct this by telling the interface card before graphics is started not to force a <CR>. It is possible to do this with most interface cards (if not all), but the method of doing so varies depending on the card.

Another important consideration is making sure the interface card does not intercept some of the characters being sent to it

as commands to the interface. For example many interface cards use the ASCII code ^I (control I, decimal 9, hex 9) as a command character. If this happens, some of the bytes intended to be sent to the printer never get passed the interface card.

If some of the problems mentioned above apply to the interface card you are using, one solution is writing your own machine language subroutine to send characters to the printer. In most cases this is not necessary because most intelligent interface cards will usually allow the user to control whether a <CR> should be forced, and allow command characters it recognizes to be changed by the user. It is a good idea for the user to know these details about the interface card he/she is using. This information can be obtained from the interface cards manual.

2.3) CHANGING THE FONT TYPE/CHARACTER SIZE

The PC-8023A-C has the capability of printing the selected character set in several different manners. They are:

- 1) PICA (10 CPI)
- 2) ELITE (12 CPI)
- 3) CONDENSED (17 CPI)
- 4) PROPORTIONAL

Using these font types is described in subsections of section 2.3. In addition to the font type, two other features the printer offers which play a role in the looks of the printed characters, are character enlargement (see section 2.4) and character enhancement (see section 2.5).

Below is the alphabet printed in the four different fonts listed above:

PICA

ABCDEFGHIJKLMNOPQRSTUVWXYZ

ELITE

ABCDEFGHIJKLMNOPQRSTUVWXYZ

CONDENSED

ABCDEFGHIJKLMNOPQRSTUVWXYZ

PROPORTIONAL

ABCDEFGHIJKLMNOPQRSTUVWXYZ

2.3.1) PICA CHARACTERS

To change the font type to pica (10 characters per inch), use the escape sequence below:

ASCII: ESC N

DECIMAL: 27 78

HEX: 1B 4E

==> FROM APPLESOFT: PRINT CHR\$(27);"N";

When the printer is powered on this mode is automatically selected (this is only true if switch SW2-5 is OFF, otherwise proportional is default). Because it is default the above escape sequence is only needed to reselect pica if it has been previously deselected by another font type.

2.3.2) ELITE CHARACTERS

To change the font type to elite (12 characters per inch), use the escape sequence below:

ASCII: ESC E

DECIMAL: 27 69

HEX: 1B 45

==> FROM APPLESOFT: PRINT CHR\$(27);"E";

NOTE

The manual states "In this mode [elite], dot-image printing cannot be performed". However, dot-image printing can be performed in this mode, with more dots per line (768 dots instead of 640).

2.3.3) CONDENSED CHARACTERS

To change the font type to condensed (17 characters per inch), use the escape sequence below:

ASCII: ESC Q

DECIMAL: 27 81

HEX: 1B 51

==> FROM APPLESOFT: PRINT CHR\$(27);"Q";

NOTE

The manual states "in this mode [condensed], enhanced printing and dot-image graphics printing cannot be performed". However, both can be performed from this mode, with an increase in dots per line in the dot-image graphics mode (1088 dots instead of 640).

2.3.4) PROPORTIONAL CHARACTERS

To change the font type to proportional, use the escape sequence below:

ASCII: ESC P

DECIMAL: 27 80

HEX: 1B 50

==> FROM APPLESOFT: PRINT CHR\$(27);"P";

NOTE

The printer allows for n-dot spacing while in the proportionally spaced printing mode (where n is from 1 to 6). This simply means that it is possible to widen the spaces between characters printed in proportional mode. See section 2.6 for more information.

2.4) PRINTING ENLARGED CHARACTERS

The printer is capable of enlarging any of the font types, pica, elite, condensed, or proportional. The four font types are printed enlarged below:

PICA-ENLARGED

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

ELITE-ENLARGED

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

CONDENSED-ENLARGED

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

PROPORTIONAL-ENLARGED

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

To print enlarged characters in JA7 mode use the control code given below:

ASCII: ^R (DC2)

DECIMAL: 18

HEX: 12

==> FROM APPLESOFT: PRINT CHR\$(18);

To take the printer out of the enlarged character mode while the printer is in JA7 mode the control code given below must be sent to the printer:

ASCII: ^T (DC4)

DECIMAL: 20

HEX: 14

==> FROM APPLESOFT: PRINT CHR\$(20);

(The above two control codes are for when the printer is in JA7 mode (SW1-1,2,3: OFF, and SW2-6: ON). If the printer is NOT in JA7 mode ^N enlarges printing and ^O deselects enlarged characters.) ¹⁴

¹⁵
dec 15

EXAMPLE 2.4

```

1 REM EXAMPLE 2.4 BY JAY ZIPNICK
2 REM PRINTER INTERFACE IN SLOT 1
10 PR#1
20 REM
30 REM PRINT ALPHABET REGULAR AND ENLARGED
40 REM
50 PRINT "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
60 REM
70 REM ----- TURN ON CHARACTER ENLARGEMENT -----
80 REM
90 PRINT CHR$(18);
100 PRINT "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
110 REM
120 REM ----- TURN OFF CHARACTER ENLARGEMENT -----
130 REM
140 PRINT CHR$(20);
150 PR#0
160 END

```

The output of the above program is printed below:

```

ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ

```

2.5) CHARACTER ENHANCEMENT

The output of the printer can be enhanced for better quality print outs. (The way this works is by printing a double horizontal dot instead of a single dot. This extra dot has a slight horizontal offset so that the gaps between pairs of horizontal dots are filled.)

To select enhanced character mode the escape sequence shown below should be used:

ASCII: ESC !

DECIMAL: 27 33

HEX: 1B 21

==> FROM APPLESOFT: PRINT CHR\$(27);"!";

To deselect enhanced mode use the following escape sequence:

ASCII: ESC "

DECIMAL: 27 34

HEX: 1B 22

==> FROM APPLESOFT: PRINT CHR\$(27);CHR\$(34);

Because the quality of the printout is improved when enhanced printing mode is used, it is often desirable to use this mode in final copies of reports, letters, memos, etc.... However during preliminary drafts this is not necessary, and in fact is not wanted because it will wear out the ribbon faster. Therefore after a text is determined to be a final copy, the escape sequence <ESC> ! could be added to enhance the final print out.

Although the printer can print condensed characters enhanced (despite the manual saying this can't be performed), printing in this mode (condensed-enhanced) does little, if anything at all to improve the print quality. Because the dots are so close together in condensed mode, enhancement might even degrade the print quality instead of improve it. It is left to the user to print condensed-enhanced and condensed-unenhanced and see which he/she prefers.

Below is an example of enhanced pica versus unenhanced pica:

This sentence is not enhanced.
This sentence is enhanced.

2.6) INTER-CHARACTER SPACING FOR PROPORTIONALLY SPACED PRINTING (DOT SPACING)

In the proportionally spaced printing mode, additional spaces having lengths from 1 to 6 horizontal dots may be added between characters. This feature is for proportionally spaced printing only. To add n-dot spaces between two characters, execute the following escape sequence:

ASCII: ESC n (where $1 \leq n \leq 6$)

DECIMAL: 27 1-6

HEX: 1B 1-6

==> FROM APPLESOFT: PRINT CHR\$(27);CHR\$(n);

It is important to note that this second character, n, is not the ASCII characters "1" to "6". This second character (byte) is numerically in the range 1 to 6, and corresponds to the control characters ^A to ^F.

EXAMPLE 2.6

```
10 PR#1
20 REM  SELECT PROPORTIONAL CHARACTERS
30 REM
40 PRINT CHR$(27);"P";
50 REM
60 REM  ADD 6 DOT SPACES BETWEEN "ABC" AND "DEF"
70 REM  THEN PRINT THE TWO STRINGS WITHOUT THE SEPARATION
80 REM
90 PRINT "ABC";
100 PRINT CHR$(27);CHR$(6);
110 PRINT "DEF"
120 PRINT "ABCDEF"
140 REM  RESTORE PICA
150 PRINT CHR$(27);"N"
160 PR#0
170 END
```

The output of this program is printed below:

```
ABC DEF
ABCDEF
```

2.7) UNDERLINE PRINTING

Underline printing is a mode in which the lower, ninth dot is printed. Anything printed can be underlined. This means the printable ASCII characters, CG graphics characters, the Greek character set, Katakana, and even dot image graphics can be underlined.

Underlining can be turned on by using the escape sequence below:

ASCII: ESC X

DECIMAL: 27 88

HEX: 1B 58

==> FROM APPLESOFT: PRINT CHR\$(27);"X";

To turn off underlining use the escape sequence below:

ASCII: ESC Y

DECIMAL: 27 89

HEX: 1B 59

==> FROM APPLESOFT: PRINT CHR\$(27);"Y";

EXAMPLE 2.7

The Applesoft program below will print the alphabet underlined with every capital letter doubled by its lowercase equivalent.

```

1 REM EXAMPLE 2.7 BY JAY ZIPNICK
2 REM PRINTER INTERFACE IN SLOT 1
10 PR#1
20 REM
30 REM  TURN ON UNDERLINING
40 REM
50 E$=CHR$(27)
60 PRINT E$;"X";
70 REM
80 REM  PRINT UPPER AND LOWERCASE ALPHABET
90 REM  (LOWER ALPHABET IS 32 CHARACTERS AFTER UPPER CASE ONE)
100 REM
110 FOR L=ASC("A") TO ASC("Z")
120 PRINT CHR$(L);CHR$(L+32);
130 NEXT L
140 REM
150 REM  TURN OFF UNDERLINING AND FORCE A (CR) TO OUTPUT BUFFER
160 REM
170 PRINT E$;"Y"
180 PR#0
190 END

```

The output of this program is printed below:

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

2.8) SETTING THE LEFT MARGIN

The printer allows the left margin to be set any number of spaces to the right of the default (absolute) left margin. The left margin is normally set to zero but this can be changed through the escape sequence given below:

ASCII: ESC L n1n2n3

DECIMAL: 27 76 d1d2d3

HEX: 1B 4C h1h2h3

==> FROM APPLESOFT: PRINT CHR\$(27);"L";"n1n2n3";

The three characters after the <escape> "L" are THREE DECIMAL DIGITS EXPRESSED IN ASCII which is the number of spaces from the absolute left of the line, that the left margin will start at. For example the Applesoft statement

PRINT CHR\$(27)"L015";

will set the left margin 15 spaces from the absolute left.

Although the number of spaces is defined in the escape sequence, the size of the spaces is not, and is a function of the printing mode at the time of the setting of the left margin. For example, if the left margin is set for seven spaces when in condensed mode, this means that the left margin in subsequent printing will be seven CONDENSED spaces from the absolute left.

NOTE

The manual states that "in the proportional printing mode, the left margin set command cannot be used". However this is not correct, setting the left margin while the printer is in proportional mode is the same as setting the left margin while the printer is in pica. The spaces from the absolute left are pica spaces.

2.9) HORIZONTAL TABULATION

This section describes how to set and clear horizontal tab positions and how to use the tabs once they are set. Setting and clearing horizontal tabs serve the same function as they do on ordinary typewriters.

To set horizontal tabs use the escape sequence given below:

ASCII: ESC ($\alpha, \beta, \dots, \xi$.

DECIMAL: 27 40 $\alpha, \beta, \dots, \xi$.

HEX: 1B 28 $\alpha, \beta, \dots, \xi$.

==> FROM APPLESOFT: PRINT CHR\$(27);"(";" $\alpha, \beta, \dots, \xi$."

To clear horizontal tabs at specified positions use the escape sequence given below:

ASCII: ESC) $\alpha, \beta, \dots, \xi$.

DECIMAL: 27 41 $\alpha, \beta, \dots, \xi$.

HEX: 1B 29 $\alpha, \beta, \dots, \xi$.

==> FROM APPLESOFT: PRINT CHR\$(27);")";" $\alpha, \beta, \dots, \xi$."

WHERE

$\alpha \beta \xi$: 3-digit decimal numbers
 ,: Continuation code
 .: Completion code

To clear all horizontal tab positions use the escape sequence below:

ASCII: ESC 2

DECIMAL: 27 50

HEX: 1B 32

==> FROM APPLESOFT: PRINT CHR\$(27);"2";

After tabs have been set, sending a horizontal tab character to the printer will move the print head over to the next horizontal position where a tab is set. If no tabs are set sending a tab character to the printer has no effect.

To move the print head to the next tab position which is set, use the control character below:

ASCII: ^I (HT)

DECIMAL: 9

HEX: 09

==> FROM APPLESOFT: PRINT CHR\$(9);

EXAMPLE 2.9

- The following Applesoft statement will set tabs at columns 10, 20, 30, and 40:

```
PRINT CHR$(27);"(";"010,020,030,040.";
```

- The following Applesoft statement will clear the tabs set at columns 20 and 40:

```
PRINT CHR$(27);")";"020,040.";
```

- If the above two statements were executed tabs would now be set at columns 10 and 30. The Applesoft statement below will use these tab positions.

```
PRINT CHR$(9);"COL10";CHR$(9);"COL30"
```

The output appears below:

COL10

COL30

1234567890123456789012345678901234567890 . . . (columns)

- The following Applesoft statement will clear all tabs.

```
PRINT CHR$(27);"2";
```

NOTE

The maximum number of horizontal tab positions allowed is 16. If an error occurs while setting the horizontal tab positions (i.e. leaving out a comma between tab positions), all the horizontal tab positions are cleared (not just the ones you were attempting to set). Although the manual states "this function is not in effect in proportional printing mode", this is not correct. Setting tabs in proportional printing mode will set them up as if they were set in the pica printing mode.

Many intelligent printer interfaces for the Apple use ^I (the horizontal tab character) for a special command character to send information to the interface card for printer control. If the interface card being used with the NEC PC-8023A-C printer uses ^I as a command character, you must change it to another character to send a horizontal tab to the printer, otherwise the interface card will intercept the ^I as the start of a command. It should be possible to change the interface command character from ^I to another control character with most interface cards. The method of doing this depends on the interface card being used, therefore you must consult the documentation for your particular card.

2.10) LINE SPACING FOR N LINES

The printer is capable of spacing down N lines where N is between 0 and 15 inclusive, by sending the printer a two character control sequence given below:

ASCII: US 16+n (where 0 ≤ n ≤ 15)

DECIMAL: 31 16-31

HEX: 1F 10-1F

==> FROM APPLESOFT: PRINT CHR\$(31);CHR\$(16+n);

(The most significant three bits of the second byte are irrelevant.)

EXAMPLE 2.10

To space down 10 lines use the following Applesoft statement:

PRINT CHR\$(31);CHR\$(26);

2.11) VERTICAL TABULATION

This section describes the Vertical Form Unit (VFU). The VFU provides a vertical tab function. For the PC-8023A-C, the form length is considered to be 66 lines (72 if DIP switch SW1-4 is ON). The first line is called TOF (Top of Form), and the last line printed is called the BOTTOM. When the BOTTOM position is sensed while printing, the printer automatically feeds the form to the next TOF position. This function allows the bottom margin to be set. (The BOTTOM is set at only one position per page.)

By setting the VFU, vertical tab positions can be set at arbitrary lines between the TOF and BOTTOM. When the vertical tab code is sent from the Apple, the form in the printer is sent to the next vertical tab position. The VFU has a 6-channel memory in which vertical tab positions (including the TOF) can be stored. These six channels are denoted by CH1, CH2, ... and CH6. CH1 is used for the TOF position and both CH1 and CH2 for the BOTTOM position (see figure 2.11a). In CH2, vertical tabs are set at 6-line intervals for the initial values when the power is turned on. Therefore when the VT code (Vertical Tab, 11 decimal, 0B hex) is sent to the printer, the form is fed to the vertical tab positions set by CH2. If the FF code (Form Feed, 12 decimal, 0C hex) is sent to the printer, the form is fed to the vertical tab position set by CH1, i.e., the TOF position. Users can freely set their own vertical tab positions between CH2 and CH6.

To perform the vertical tab function for CHn, use the control sequence below:

ASCII: US n (where $1 \leq n \leq 6$)

DECIMAL: 31 1-6

HEX: 1F 01-06

==> FROM APPLESOFT: PRINT CHR\$(31);CHR\$(n);

(The three most significant bits of the second byte of this code are irrelevant. If the fourth bit is a one then the printer command is used for line feeding N lines where N is specified by the last four bits (see section 2.10). Either way the ASCII US character is used as a vertical forms motion command by the printer.)

The ASCII VT (vertical tab) control character can be used alone to refer to the next vertical tabulation position specified in channel 2. Note that the VT command (below) will always use channel 2.

ASCII: ^K (VT)

DECIMAL: 11

HEX: 0B

==> FROM APPLESOFT: PRINT CHR\$(9);

The VFU can be considered to be a continuous 6-level tape if viewed graphically as in the figure below.

```

      C      C C
      H . . . H H
      6      2 1

0 0 0 0 0 0

0 0 0 0 0 0 • ← TOF

0 0 0 0 0 0      2nd line
      .
      .
      .
      .
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0 • ← Vertical tabulation
                  position of CH2
0 0 0 0 0 0
0 0 0 0 0 0 • • ← BOTTOM
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0      66th line
0 0 0 0 0 0 • ← TOF

```

FIGURE 2.11a

SUMMARY OF THE VFU

- (1) Max. form length (between TOFs):
 - 66 lines (72 lines if SW1-4 is ON)
- (2) Initialization format:
 - When the VFU is not set externally, it is automatically set as follows.
 - Form length: 66 lines.
 - Tab positions: Set in CH2 at every 6-line interval from TOF.
- (3) Setting the TOF position:
 - Only CH1 is used to set the TOF position. All other channels are zero.
- (4) Setting the VT positions:
 - CH2 through CH6 are used to set vertical tab positions. (Vertical tabbing to these positions is accomplished via the <VT> and <US> commands explained above.)
- (5) Setting the BOTTOM position:

- CH1 and CH2 are used at the same time to set the BOTTOM position, and they are set to only one position within the form.

(6) Data code:

- Every byte of data used to set the VFU must be accompanied by a second byte as follows:

Bit	7	6	5	4	3	2	1	0
2-byte data format	< x	1	CH6	CH5	CH4	CH3	CH2	CH1
	< x	1	x	x	x	x	x	x

x: Irrelevant

Each position where CHn has a logic "1" is a vertical tab position for channel n. Even though only the first byte contains data for setting the VFU (as explained in (7) below), a second byte with a one in bit 6 must also be sent to the printer.

(7) The following codes must be input to the printer to set the VFU.

[illegible]

FIGURE 2.11b

LOADING THE VFU

- To load the VFU the first character (start character) must be the ASCII control character GS (Group Separator, decimal 29, hex 1D). (See figure 2.11b and the program example below).
- Next send 66 pairs of bytes, one pair for each line on the form. These data bytes represent which of the six channels is to be set for each line. The format of these bytes is given above in paragraph (6) "data code" in the above "SUMMARY OF THE VFU". Channels 2 through 6 are vertical tab channels and may be configured in a combination the user desires, as long as channel 1 contains a zero (remember channel one contains a "1" only if it indicates a TOF or BOTTOM).
- After the 66 byte pairs have been entered a 67th byte pair follows containing a TOF code.

If an error occurs while loading the VFU (i.e. not specifying TOF immediately after the start command) the printer will default to the pre-programmed format.

EXAMPLE 2.11

The example below sets the printer for a 10-character left margin, proportional printing and automatic page ejection every 60 lines.

```

1 REM EXAMPLE 2.11
10 PR#1
20 E$=CHR$(27)
30 REM
40 REM SET LEFT MARGIN AND PROPORTIONAL PRINTING
50 REM
60 PRINT E$;"L";"010";
70 PRINT E$;"P";
80 REM
90 REM SET VFU
100 REM
110 PRINT CHR$(29);"A";
120 FOR N=1 TO 117
130 PRINT "@";
140 NEXT N
150 PRINT "CCCCCCCCCCCCCCCCCA@";
160 PRINT CHR$(30)
170 PRINT CHR$(11)
180 PR#0
190 END

```

2.12) PROGRAMMABLE LINE SPACING

The PC-8023A-C offers three different commands for controlling line spacing. The first one sets the line spacing for 1/6 of an inch (this is default). The second one sets the line spacing for 1/8 of an inch. The third one sets the line spacing for $n/144$ of an inch, where n is between 0 and 99 inclusive and is provided by the user. These three commands are all in the form of escape sequences provided below.

Use the following escape sequence to set the line spacing for 1/6 of an inch (6 lines per inch):

ASCII: ESC A

DECIMAL: 27 65

HEX: 1B 41

==> FROM APPLESOFT: PRINT CHR\$(27);"A";

Use the following escape sequence to set the line spacing for 1/8 of an inch (8 lines per inch):

ASCII: ESC B

DECIMAL: 27 66

HEX: 1B 42

==> FROM APPLESOFT: PRINT CHR\$(27);"B";

Use the following escape sequence to set the line spacing for $n/144$ of an inch:

ASCII: ESC T n_1n_2 (where $00 \leq n_1n_2 \leq 99$)

DECIMAL: 27 84 d_1d_2

HEX: 1B 54 h_1h_2

==> FROM APPLESOFT: PRINT CHR\$(27);"T";" n_1n_2 ";

The two characters after the ESC "T" are TWO DECIMAL DIGITS EXPRESSED IN ASCII which is the number of 1/144 of an inch to move up the paper. The width of one dot is 2/144".

EXAMPLE 2.12

The following Applesoft statement will set the line spacing for 16/144".

PRINT CHR\$(27);"T";"16";

2.13) CHANGING THE LINE FEED DIRECTION

The PC-8023A-C can move the paper both up and down.

To have the line feed occur in the forward direction (default) use the escape sequence below:

ASCII: ESC f (note lowercase f)

DECIMAL: 27 102

HEX: 1B 66

==> FROM APPLESOFT: PRINT CHR\$(27);CHR\$(102);

To have the line feed occur in the reverse direction use the escape sequence below:

ASCII: ESC r (note lower case r)

DECIMAL: 27 114

HEX: 1B 72

==> FROM APPLESOFT: PRINT CHR\$(27);CHR\$(114);

2.14) INCREMENTAL MODE/LOGICAL-SEEKING MODE SELECTION

The printer can be in a logical-seeking mode (default), or incremental mode. In logical-seeking mode the printer chooses the direction to move the print head so that the print head moves the shortest distance. In incremental mode, the printer prints each character without trying to determine the shortest path for the print head.

There are minor differences when the printer is in incremental mode. Printing in logical-seeking mode is faster than printing in the incremental mode. When a carriage return is received in incremental mode the print head automatically moves to the left margin. Printing in the logical-seeking mode ignores backspaces, so that backspacing can only be performed when the printer is in the incremental mode. When the printer is in the logical-seeking mode, printed lines can be misaligned at the left margin. That is, every other line is offset by one dot horizontally. This normally can't be noticed when outputting text. In the dot-image graphics mode, this might be noticeable in multi-line pictures by someone with a sharp eye. If the printer is put in incremental mode, this can be corrected.

To place the printer in incremental mode use the escape sequence below:

ASCII: ESC [

DECIMAL: 27 91

HEX: 1B 5B

==> FROM APPLESOFT: PRINT CHR\$(27);CHR\$(91);

To place the printer in logic seeking mode use the escape sequence below:

ASCII: ESC]

DECIMAL: 27 93

HEX: 1B 5D

==> FROM APPLESOFT: PRINT CHR\$(27);CHR\$(93);

2.15) BACKSPACING

Backspacing moves the print head back one character unless it is already at the left margin, in which case the print head cannot be backspaced any further. Backspacing will only work if the printer is in incremental mode (see section 2.14).

To backspace, use the control code below:

ASCII: ^H (BS) (This only works in incremental mode.)

DECIMAL: 8

HEX: 08

==> FROM APPLESOFT: PRINT CHR\$(8);

2.16) PRINTER SELECT AND DESELECT MODE

The printer can be selected (placed on-line) and deselected (taken off-line) by sending the right control characters to the printer. DIP-switch SW1-5 must be in the OFF (open) position to process these control characters as select/deselect commands (in the ON position these characters are ignored).

To select the printer use the control character given below:

ASCII: ^Q (DC1) (valid only if SW1-5 if OFF)

DECIMAL: 17

HEX: 11

==> FROM APPLESOFT: PRINT CHR\$(17);

In the PE (paper empty) state, the select code (DC1) is invalid.

To deselect the printer use the control character given below:

ASCII: ^S (DC3) (valid only if SW1-5 if OFF)

DECIMAL: 19

HEX: 13

==> FROM APPLESOFT: PRINT CHR\$(19);

Normally SW2-2 should be OFF. If DIP-switch SW2-2 is ON then it is not only necessary to have the printer selected, but it is also necessary to select it via the address determined by SW2-3,4 by using an escape sequence. One way of looking at this two level selection process is as follows. The first level of selection puts the printer ON-LINE (the select light goes on). The second level of selection, addresses the printer and makes it an ACTIVE device. If the device is on-line but not active it will receive all the characters but not print them until it is selected.

If SW2-2 is ON this means the printer has its own "address" determined by SW2-3,4. When the printer is given an address (SW2-2 is ON), the printer must be selected via this address by using the escape sequence below:

ASCII: ESC a-d

DECIMAL: 27 97-100

HEX: 1B 61-64

==> FROM APPLESOFT: PRINT CHR\$(97+n);

Above the n represents the device number set by SW2-3,4 which ranges from 0 to 3 inclusive. The device number, n, is determined as follows:

DEVICE NUMBER	SW2-3	SW2-4
0	OFF	OFF
1	ON	OFF
2	OFF	ON
3	ON	ON

Setting the device address can be useful if you have more than one printer being driven from the same port and want to select which one will print.

If SW2-2 is ON then the following escape sequence will release the devices from the selected state.

```
ASCII:      ESC
DECIMAL:    27  96
HEX:       1B  60
==> FROM APPLESOFT:  PRINT CHR$(96);
```

2.17) THE CANCEL DATA COMMAND

In logic seeking mode the data since the last carriage return or line feed that has not been printed can be cancelled.

The control code to cancel data is provided below:

```
ASCII:      ^X  (CAN)  (valid in logic
                        seeking mode only)
DECIMAL:    24
HEX:       18
==> FROM APPLESOFT:  PRINT CHR$(19);
```

All the control codes received before receiving this code are valid and the printer mode is changed to the last mode received before receiving the CAN code.

2.18) THE LINE FEED COMMAND

The printer will issue a line feed when it receives the ASCII line feed control character shown below.

```
ASCII:      ^J  (LF)
DECIMAL:    10
HEX:       0A
==> FROM APPLESOFT:  PRINT CHR$(10);
```

2.19) THE FORM FEED COMMAND

The printer will issue a form feed when it receives the ASCII form feed control character shown below.

ASCII: ^L (FF)

DECIMAL: 12

HEX: 0C

==> FROM APPLESOFT: PRINT CHR\$(12);

The form feed control character will move the paper to the top of the next page. The NEC-8023A-C will not loose track of where the top of form should be (unlike some of its competitors) even if the programmable line spacing has been changed. The printer will always respond to the form feed character in incremental mode. In the logical-seeking mode it will respond to a form feed if it is the first printable character of a line, but it will not respond to the form feed if it is in the middle of a line.

3) APPLESOFT PRINTER TEST PROGRAM

The program described in the following pages includes all the functions of the printer. The practical use of each function described in section 2 can be better understood by actually using this program or by reading the listing.

```

1  REM *****
2  REM *
3  REM * PRINTER TEST PROGRAM *
4  REM *
5  REM *TRANSLATED TO APPLESOFT*
6  REM *
7  REM *   BY JAY ZIPNICK   *
8  REM *
9  REM *****
10 REM
20 REM PRINTER ASSUMED TO BE IN JA7 MODE
30 REM
40 PR# 1: REM OUTPUT TO PRINTER
50 GOSUB 60000: REM INITIALIZE INTERFACE CARD
60 REM
70 REM ***** THROUGHOUT E$ = <ESCAPE> *****
80 REM
90 E$ = CHR$ (27)
100 REM
110 REM =====
120 REM ***** PRINT ALL CHARACTERS IN EACH FONT *****
130 REM =====
140 REM
150 S$ = " ": GOSUB 10000
160 REM
170 REM =====
180 REM ***** PRINT ALL CHARACTERS IN EACH FONT ENHANCED *****
190 REM =====
200 REM
210 REM ----- TURN ON CHARACTER ENHANCEMENT -----
220 REM
230 PRINT E$;"!";
240 S$ = "ENHANCED ": GOSUB 10000
250 GOSUB 10000
260 REM
270 REM ----- TURN OFF CHARACTER ENHANCEMENT -----
280 REM
290 PRINT E$; CHR$ (34);
300 REM
310 REM =====
320 REM ***** PRINT ALL CHARACTERS IN EACH FONT ENLARGED *****
330 REM =====
340 REM
350 REM ----- TURN ON CHARACTER ENLARGEMENT -----
360 REM
370 PRINT CHR$ (18);
380 S$ = "ENLARGED ": GOSUB 10000
390 REM
400 REM ----- TURN OFF CHARACTER ENLARGEMENT -----
410 REM
420 PRINT CHR$ (20);
430 REM
440 REM =====
450 REM ***** PRINT ALL CHARACTERS IN EACH FONT ENLARGED AND ENHANCED *****
460 REM =====
470 REM
480 REM ----- ENLARGE AND ENHANCE -----
490 REM

```

```

500 PRINT CHR$(18);E$;"!";
510 S$ = "ENLARGED & ENHANCED ": GOSUB 10000
520 REM
530 REM ----- TURN OFF ENLARGEMENT AND ENHANCEMENT -----
540 REM
550 PRINT CHR$(20);E$; CHR$(34);
560 REM
570 REM
580 REM =====
590 REM ***** DOT SPACING *****
600 REM =====
610 REM
620 PRINT E$;"P";: REM SELECT PROPORTIONAL
630 PRINT
640 PRINT "PROPORTIONAL CHARACTERS"
650 PRINT "DOT SPACING"
660 REM AJ$ = LOWER CASE ALPHABET A-J
670 REM KZ$ = LOWER CASE ALPHABET K-Z
680 REM AZ$ = LOWER CASE ALPHABET
690 AJ$ = "";KZ$ = ""
700 FOR I = ASC("A") TO ASC("J")
710 AJ$ = AJ$ + CHR$(I + 32)
720 NEXT I
730 FOR I = ASC("K") TO ASC("Z")
740 KZ$ = KZ$ + CHR$(I + 32)
750 NEXT I
760 REM
770 AZ$ = AJ$ + KZ$
780 PRINT AZ$
790 FOR N = 1 TO 6
800 PRINT AJ$;
810 REM
820 REM ----- DOT SPACING (N DOTS) 1<=N<=6 -----
830 REM
840 PRINT E$; CHR$(N);
850 REM
860 PRINT KZ$;". DOT SPACING (";N;" DOTS) BETWEEN J AND K."
870 NEXT N
880 PRINT E$;"N";: REM RESELECT PICA
890 REM
900 REM
910 REM =====
920 REM ***** CANCEL TEST *****
930 REM =====
940 REM
950 PRINT
960 PRINT "CANCEL TEST"
970 PRINT "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
980 PRINT " THIS IS TEXT."
990 PRINT "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
1000 REM
1010 REM ----- CANCEL CODE -----
1020 REM
1030 PRINT CHR$(24);
1040 REM THE SECOND ALPHABET SHOULD NOT BE PRINTED.
1050 REM THE PRINT STATEMENT BELOW WILL BE PRINTED.
1060 PRINT " THIS IS TEXT."
1070 PRINT : PRINT
1080 REM
1090 REM

```

```

1100 REM =====
1110 REM ***** UNDERLINING *****
1120 REM =====
1130 REM
1140 PRINT "THIS IS ";
1150 REM
1160 REM ----- START UNDERLINING -----
1170 REM
1180 PRINT E$;"X";
1190 REM
1200 PRINT "UNDERLINED";
1210 REM
1220 REM ----- STOP UNDERLINING -----
1230 REM
1240 PRINT E$;"Y";
1250 REM
1260 PRINT ". THIS IS NOT UNDERLINED."
1270 PRINT : PRINT
1280 REM
1290 REM
1300 REM =====
1310 REM ***** VARIABLE LINE SPACING *****
1320 REM =====
1330 REM
1340 PRINT
1350 PRINT "VARIABLE LINE SPACING"
1360 PRINT
1370 REM AA$ = LOWER CASE ALPHABET, SPACE, AND UPPER CASE ALPHABET
1380 AA$ = AZ$ + " ABCDEFGHIJKLMNOPQRSTUVWXYZ"
1390 REM
1400 REM ----- LINE SPACING (1/6 INCH) -----
1410 REM
1420 PRINT E$;"A";
1430 REM
1440 PRINT "LINE SPACING (1/6 INCH)"
1450 FOR L = 1 TO 2: PRINT AA$: NEXT L
1460 PRINT
1470 REM
1480 REM ----- LINE SPACING (1/8 INCH) -----
1490 REM
1500 PRINT E$;"B";
1510 REM
1520 PRINT "LINE SPACING (1/8 INCH)"
1530 FOR L = 1 TO 2: PRINT AA$: NEXT L
1540 PRINT
1550 REM
1560 REM ----- LINE SPACING (16/144 INCH) -----
1570 REM
1580 PRINT E$;"T";"16";
1590 PRINT "LINE SPACING (16/144 INCH)"
1600 FOR L = 1 TO 2: PRINT AA$: NEXT L
1610 PRINT
1620 REM
1630 REM
1640 FOR N = 20 TO 99 STEP 5
1650 PRINT E$;"T";STR$(N);
1660 PRINT "LINE SPACING ("&N&"/144 INCH)"
1670 FOR L = 1 TO 2: PRINT AA$: NEXT L
1680 PRINT : PRINT
1690 NEXT N

```



```

1700 PRINT E$;"A";: REM RESTORE 1/6 INCH LINE SPACING
1710 PRINT : PRINT
1720 REM
1730 REM
1740 REM =====
1750 REM ***** USING HORIZONTAL TABS *****
1760 REM =====
1770 REM
1780 PRINT "HORIZONTAL TABULATION"
1790 REM
1800 REM ----- SET HORIZONTAL TABS -----
1810 REM
1820 PRINT E$;"(";"010,020,030,040,050.";
1830 T$ = CHR$ (9): REM TAB CHAR
1840 PRINT "X";T$;"TAB1";T$;"TAB2";T$;"TAB3";T$;"TAB4";T$;"TAB5"
1850 REM
1860 REM ----- CLEAR HORIZONTAL TABS -----
1870 REM
1880 PRINT E$;")";"020,040.";
1890 PRINT "X";T$;"TAB1";T$;"TAB2";T$;"TAB3"
1900 REM
1910 REM ----- CLEAR ALL HORIZONTAL TABS -----
1920 REM
1930 PRINT E$;"2";
1940 PRINT "X";T$;"TAB1";T$;"TAB2";T$;"TAB3"
1950 REM
1960 REM
1970 REM =====
1980 REM ***** MULTIPLE LINE FEEDS *****
1990 REM =====
2000 REM
2010 PRINT
2020 US$ = CHR$ (31)
2030 PRINT "15 LINE FEEDS FOLLOW"
2040 REM
2050 REM ----- MULTIPLE (N) LINE FEEDS -----
2060 REM
2070 PRINT
2080 N = 15
2090 PRINT US$; CHR$ (16 + N);
2100 PRINT "14 LINE FEEDS FOLLOW"
2110 N = 14
2120 PRINT US$; CHR$ (16 + N);
2130 PRINT "1 LINE FEED FOLLOWS"
2140 N = 1
2150 PRINT US$; CHR$ (16 + N);
2160 REM
2170 REM
2180 REM =====
2190 REM ***** SETTING THE LEFT MARGIN *****
2200 REM =====
2210 REM
2220 PRINT
2230 REM
2240 REM ----- SETTING LEFT MARGIN -----
2250 REM
2260 PRINT E$;"L";"010";
2270 PRINT "LEFT MARGIN 10"
2280 GOSUB 11000: REM OUTPUT ALL CHARACTERS
2290 PRINT

```

in MSU
use TAB function

```

2300 PRINT E$;"L";"020";
2310 PRINT "LEFT MARGIN 20"
2320 GOSUB 11000: REM OUTPUT ALL CHARACTERS
2330 PRINT
2340 PRINT E$;"L000";: REM RESTORE DEFAULT LEFT MARGIN
2350 REM
2360 REM
2370 REM =====
2380 REM ***** DOT-IMAGE GRAPHICS *****
2390 REM =====
2400 REM
2410 PRINT "GRAPHIC PRINT"
2420 PRINT
2430 REM
2440 REM ----- PUT PRINTER IN INCREMENTAL MODE -----
2450 REM
2460 PRINT E$; CHR$ (91);
2470 REM (THIS ENSURES ALL DOTS ARE ALLIGNED VERTICALLY AT EVERY NEW LINE)
2480 REM
2490 PRINT E$;"T14";: REM LINE SPACING (14/144 INCH)
2500 FOR LINE = 1 TO 20
2510 REM
2520 REM ----- SELECT GRAPHIC MODE (255 BYTES) -----
2530 REM
2540 PRINT E$;"S";"0256";
2550 REM
2560 REM PRINT BIT PATTERNS FOR "X"
2570 REM
2580 FOR X = 127 TO 64 STEP - 1: PRINT CHR$ (X);: NEXT X
2590 FOR X = 64 TO 127: PRINT CHR$ (X);: NEXT X
2600 FOR X = 127 TO 64 STEP - 1: PRINT CHR$ (X);: NEXT X
2610 FOR X = 64 TO 127: PRINT CHR$ (X);: NEXT X
2620 PRINT
2630 NEXT LINE
2640 REM
2650 REM PRINT VERTICAL STRIPES
2660 REM
2670 FOR LINE = 1 TO 20
2680 PRINT E$;"S0600";: REM 600 BYTES OF GRAPHICS
2690 FOR N = 1 TO 150
2700 PRINT CHR$ (255); CHR$ (255); CHR$ (0); CHR$ (0);
2710 NEXT N
2720 PRINT
2730 NEXT LINE
2740 REM
2750 REM OUTPUT CHECKER BOARD PATTERN OF DOTS
2760 REM
2770 PRINT E$;"T12";: REM CHANGE LINE SPACING FOR 6 VERTICAL DOTS/LINE
2780 FOR LINE = 1 TO 20
2790 PRINT E$;"S0600";: REM 600 BYTES OF GRAPHICS
2800 FOR N = 1 TO 300
2810 PRINT CHR$ (42); CHR$ (21);
2820 NEXT N
2830 PRINT
2840 NEXT LINE
2850 REM
2860 REM ----- PUT PRINTER BACK IN LOGICAL-SEEKING MODE -----
2870 REM
2880 PRINT E$; CHR$ (93);
2890 PRINT E$;"A";: REM RESTORE DEFAULT LINE SPACING

```

```

2900 PRINT : PRINT
2910 REM
2920 REM
2930 REM =====
2940 REM ***** SETTING THE VFU *****
2950 REM =====
2960 REM
2970 REM ----- START VFU LOAD -----
2980 REM
2990 PRINT CHR$ (29);"A@";
3000 FOR L = 2 TO 66
3010 REM
3020 REM LM6 = L MOD 6
3030 REM
3040 LM6 = L - INT (L / 6) * 6
3050 REM
3060 IF LM6 = 0 THEN PRINT "B@";: REM SET CH2 VERTICAL TAB
3070 IF LM6 = 1 THEN PRINT "D@";: REM SET CH3 VERTICAL TAB
3080 IF LM6 = 2 THEN PRINT "H@";: REM SET CH4 VERTICAL TAB
3090 IF LM6 = 3 THEN PRINT "P@";: REM SET CH5 VERTICAL TAB
3100 IF LM6 = 4 THEN PRINT CHR$ (96);"@";: REM SET CH6 VERTICAL TAB
3110 IF LM6 = 5 THEN PRINT "@@";: REM SET NO VERTICAL TAB
3120 NEXT L
3130 REM
3140 REM ----- STOP VFU LOAD -----
3150 REM
3160 PRINT "A@"; CHR$ (30);
3170 REM
3180 REM
3190 REM =====
3200 REM ***** VERTICAL TABBING *****
3210 REM =====
3220 REM
3230 PRINT "VERTICAL TABULATION"
3240 FOR N = 2 TO 6
3250 REM
3260 REM ----- CHN VERTICAL TAB -----
3270 REM
3280 PRINT CHR$ (31); CHR$ (N);
3290 PRINT "CH";N;" VERTICAL TAB"
3300 NEXT N
3310 REM
3320 REM ----- TABBING VIA VERTICAL TAB CHAR < VT > -----
3330 REM
3340 PRINT CHR$ (11);
3350 PRINT "VT"
3360 REM
3370 REM ----- TOP OF FORM (TOF) -----
3380 REM
3390 PRINT CHR$ (31); CHR$ (1);
3400 REM
3410 REM
3420 PRINT "***** PRINTER TEST END *****"; PRINT
3430 PR# 0
3440 END
9900 REM x=x=x=x=x=x=x=x=x=x=x=x=x=x=x=x
9910 REM x=x ALL SUBROUTINES USED x=x
9920 REM x=x IN THIS PROGRAM x=x
9930 REM x=x APPEAR BELOW x=x
9940 REM x=x=x=x=x=x=x=x=x=x=x=x=x=x=x=x

```

```

9950 REM
9960 REM
9970 REM
9980 REM
9990 REM
10000 REM *****
10010 REM ***** PRINT CHARACTERS IN ALL FONTS *****
10020 REM *****
10030 REM
10040 REM ----- SELECT PICA CHARACTERS (10 CPI) -----
10050 REM
10060 PRINT E$;"N";
10070 PRINT S$;"PICA CHARACTERS"
10080 REM PRINT ALL CHARACTERS
10090 GOSUB 11000
10100 PRINT : PRINT
10110 REM
10120 REM ----- SELECT ELITE CHARACTERS (12 CPI) -----
10130 REM
10140 PRINT E$;"E";
10150 PRINT S$;"ELITE CHARACTERS"
10160 REM PRINT ALL CHARACTERS
10170 GOSUB 11000
10180 PRINT : PRINT
10190 REM
10200 REM ----- SELECT CONDENSED CHARACTERS (17 CPI) -----
10210 REM
10220 PRINT E$;"Q";
10230 PRINT S$;"CONDENSED CHARACTERS"
10240 REM PRINT ALL CHARACTERS
10250 GOSUB 11000
10260 PRINT : PRINT
10270 REM
10280 REM ----- SELECT PROPORTIONAL CHARACTERS -----
10290 PRINT E$;"P";
10300 PRINT S$;"PROPORTIONAL CHARACTERS"
10310 REM PRINT ALL CHARACTERS
10320 GOSUB 11000
10330 PRINT E$;"N": REM RESELECT PICA
10340 PRINT : PRINT
10350 RETURN
11000 REM *****
11010 REM ***** PRINT ALL CHARACTERS *****
11020 REM *****
11030 REM
11040 REM ----- SELECT ASCII AND PRINT -----
11050 REM
11060 PRINT E$;"$";
11070 FOR I = 32 TO 127: PRINT CHR$(I);: NEXT I
11080 REM
11090 REM ----- SELECT CG CHARACTERS AND PRINT -----
11100 REM
11110 PRINT E$;"#";
11120 FOR I = 32 TO 95: PRINT CHR$(I);: NEXT I
11130 REM
11140 REM ----- SELECT GREEK CHARACTERS AND PRINT -----
11150 REM
11160 PRINT E$;"&";
11170 FOR I = 32 TO 95: PRINT CHR$(I);: NEXT I
11180 REM

```

```

11190 REM ----- SELECT KATAKANA AND PRINT -----
11200 REM
11210 PRINT CHR$(14);
11220 FOR I = 32 TO 95: PRINT CHR$(I);: NEXT I
11230 REM
11240 REM RESELECT ASCII
11250 PRINT E$;"$";
11260 RETURN
60000 REM *****
60010 REM ***** REM INITIALIZE INTERFACE CARD *****
60020 REM *****
60030 REM
60040 REM THIS IS TO INITIALIZE THE INTERFACE CARD
60050 REM BEING USED WITH THIS TEST PROGRAM.
60060 REM THIS SUBROUTINE IS ENTIRELY DEPENDENT
60070 REM UPON THE INTERFACE CARD BEING USED.
60080 REM THIS WAS WRITTEN FOR THE EPSON APL INTERFACE CARD
60090 REM
60100 REM
60110 PRINT CHR$(9);"255N";: REM CHANGE CHARS/LINE
60120 REM
60130 REM CC = CODE FOR COMMAND CHAR.
60140 REM
60150 OLDCC = 9:CC = 7
60160 REM ***** ENTRY POINT TO CHANGE COMMAND CHAR FROM OLDCC TO VALUE 0
60170 REM
60180 PRINT CHR$(OLDCC); CHR$(CC);:OLDCC = CC
60190 RETURN

```


LINE SPACING (30/144 INCH)

abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ

LINE SPACING (35/144 INCH)

abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ

LINE SPACING (40/144 INCH)

abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ

LINE SPACING (45/144 INCH)

abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ

LINE SPACING (50/144 INCH)

abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ

LINE SPACING (55/144 INCH)

abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ

LINE SPACING (60/144 INCH)

abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ

LINE SPACING (65/144 INCH)

abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ

LINE SPACING (70/144 INCH)

abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ

LINE SPACING (75/144 INCH)

abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ

LINE SPACING (80/144 INCH)

abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ

LINE SPACING (85/144 INCH)

abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ

LINE SPACING (90/144 INCH)

abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ

LINE SPACING (95/144 INCH)

abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ

```

*          TAB1          TAB2
*          TAB1
*TAB1TAB2TAB3

```

TABLE 4

15 LINE FEEDS FOLLOW

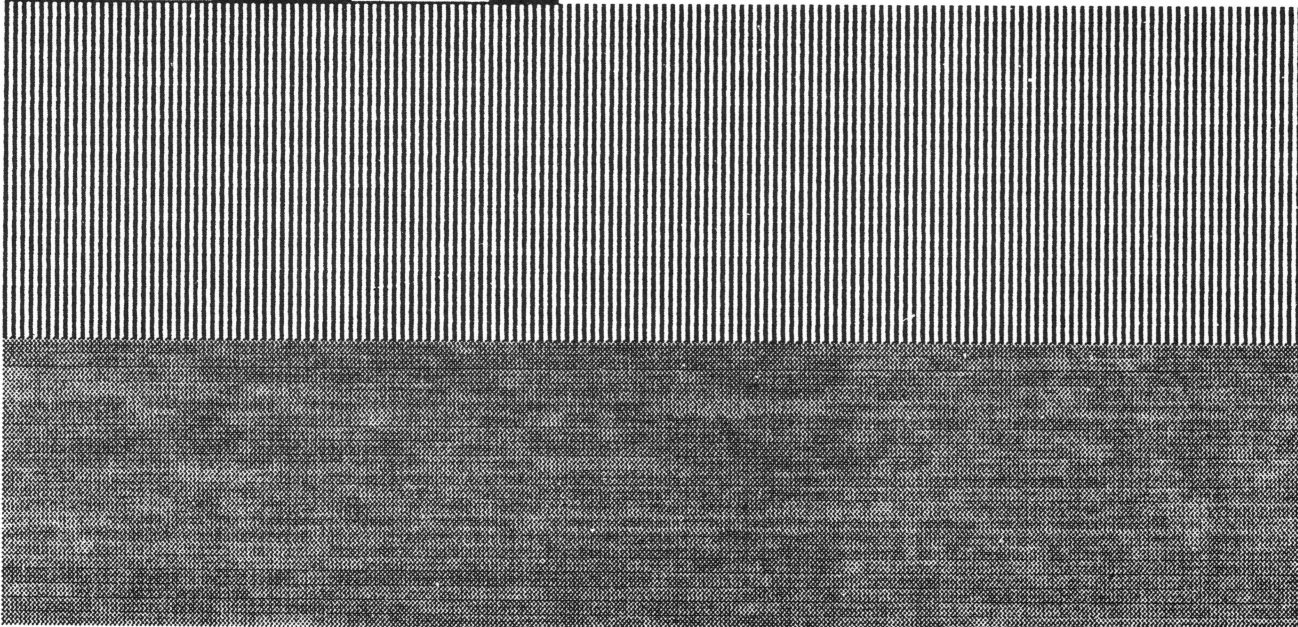
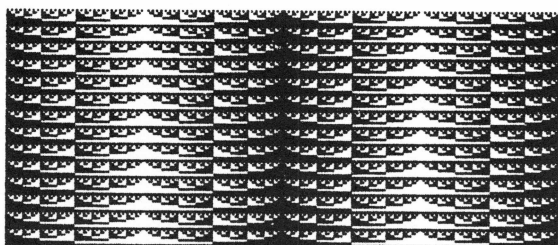
14 LINE FEEDS FOLLOW

1 LINE FEED FOLLOWS

[illegible][illegible]

GRAPHIC PRINT





VERTICAL TABULATION

CH2 VERTICAL TAB

CH3 VERTICAL TAB

CH4 VERTICAL TAB

CH5 VERTICAL TAB

CH6 VERTICAL TAB

VT

***** PRINTER TEST END *****

4) SAMPLE OUTPUT DEMONSTRATING SOME
OF THE PC-8023A-Cs CAPABILITIES

On the next page is a packing list form printed by the PC-8023A-C. This form takes advantage of many of the features the printer offers. The form is marked pointing out where different printer features are used.

FORM/TMQ-001

© 1982 by Jay Zipnick

5) SUMMARY OF ESCAPE AND FUNCTION CODES

This section provides a summary of all the escape codes and function codes available on the PC8023A-C, along with the section to refer to in this manual for detailed information.

SELECTING ASCII: (2.1.1)

ASCII	DECIMAL	HEX
<ESC> \$	27 36	1B 24

SELECTING THE CG GRAPHICS MODE: (2.1.2)

ASCII	DECIMAL	HEX
<ESC> #	27 35	1B 23

SELECTING THE GREEK MODE: (2.1.3)

ASCII	DECIMAL	HEX
<ESC> &	27 38	1B 26

SELECTING THE KATAKANA MODE: (2.1.4)

ASCII	DECIMAL	HEX
^N	14	0E

SELECTING DOT-IMAGE GRAPHICS: (2.2)

ASCII	DECIMAL	HEX
<ESC> S n1 n2 n3 n4	27 83 d1 d2 d3 d4	1B 53 h1 h2 h3 h4

SELECTING PICA CHARACTERS: (2.3.1)

ASCII	DECIMAL	HEX
<ESC> N	27 78	1B 4E

SELECTING ELITE CHARACTERS: (2.3.2)

ASCII	DECIMAL	HEX
<ESC> E	27 69	1B 45

SELECTING CONDENSED CHARACTERS: (2.3.3)

ASCII	DECIMAL	HEX
<ESC> Q	27 81	1B 51

SELECTING PROPORTIONAL CHARACTERS: (2.3.4)

ASCII	DECIMAL	HEX
<ESC> P	27 80	1B 50

CHARACTER ENLARGEMENT: (2.4)

ASCII	DECIMAL	HEX
^R	18	12

DISABLING CHARACTER ENLARGEMENT: (2.4)

ASCII	DECIMAL	HEX
^T	20	14

CHARACTER ENHANCEMENT: (2.5)

ASCII	DECIMAL	HEX
<ESC> !	27 33	1B 21

DISABLING CHARACTER ENHANCEMENT: (2.5)

ASCII	DECIMAL	HEX
<ESC> "	27 34	1B 22

DOT SPACE CONTROL: (2.6)

ASCII	DECIMAL	HEX
<ESC> ^A-^F	27 1-6	1B 1-6

UNDERLINE PRINTING: (2.7)

ASCII	DECIMAL	HEX
<ESC> X	27 88	1B 58

DISABLING UNDERLINE PRINTING: (2.7)

ASCII	DECIMAL	HEX
<ESC> Y	27 89	1B 59

SETTING THE LEFT MARGIN: (2.8)

ASCII	DECIMAL	HEX
<ESC> L n1 n2 n3	27 76 d1 d2 d3	1B 4C h1 h2 h3

SETTING HORIZONTAL TABS: (2.9)

ASCII	DECIMAL	HEX
<ESC> (t1,t2,...tn.	27 40 t1,t2,...tn.	1B 28 t1,t2,...tn.

CLEARING HORIZONTAL TABS: (2.9)

ASCII	DECIMAL	HEX
<ESC>) t1,t2,...tn.	27 41 t1,t2,...tn.	1B 29 t1,t2,...tn.

CLEARING ALL HORIZONTAL TABS: (2.9)

ASCII	DECIMAL	HEX
<ESC> 2	27 50	1B 32

HORIZONTAL TAB CHARACTER: (2.9)

ASCII	DECIMAL	HEX
^I	9	09

LINE SPACING FOR N LINES: (2.10)

ASCII	DECIMAL	HEX
<US> x x x 1 b3 b2 b1 b0 *	31 16-31	1F 10-1F

* bit pattern.

VERTICAL TABBING FOR CHn: (2.11)

ASCII	DECIMAL	HEX
<US> n	31 1-6	1F 01-06

VERTICAL TABBING FOR CH2: (2.11)

ASCII	DECIMAL	HEX
^K	11	0B

LINE SPACING FOR 6 LINES/INCH: (2.12)

ASCII	DECIMAL	HEX
<ESC> A	27 65	1B 41

LINE SPACING FOR 8 LINES/INCH: (2.12)

ASCII	DECIMAL	HEX
<ESC> B	27 66	1B 42

PROGRAMMABLE LINE SPACING FOR n/144": (2.12)

ASCII	DECIMAL	HEX
<ESC> T n1 n2	27 84 d1 d2	1B 54 h1 h2

SETTING THE LINE FEED FOR THE FORWARD DIRECTION: (2.13)

ASCII	DECIMAL	HEX
<ESC> f	27 102	1B 66

SETTING THE LINE FEED FOR THE REVERSE DIRECTION: (2.13)

ASCII	DECIMAL	HEX
<ESC> r	27 114	1B 72

PLACING THE PRINTER IN INCREMENTAL MODE: (2.14)

ASCII	DECIMAL	HEX
<ESC> I	27 91	1B 5B

PLACING THE PRINTER IN LOGICAL-SEEKING MODE: (2.14)

ASCII	DECIMAL	HEX
<ESC> J	27 93	1B 5D

PC-8023A-C SUPPLEMENT

BACKSPACING: (2.15)

ASCII	DECIMAL	HEX
^H	8	08

PRINTER SELECT: (2.16)

ASCII	DECIMAL	HEX
^Q	17	11

PRINTER DESELECT: (2.16)

ASCII	DECIMAL	HEX
^S	19	13

SELECTING PRINTER VIA DEVICE ADDRESS: (2.16)

ASCII	DECIMAL	HEX
<ESC> a-d	27 97-100	1B 61-64

RELEASE ADDRESSED DEVICES FROM SELECTED STATE: (2.16)

ASCII	DECIMAL	HEX
<ESC>	27 96	1B 60

THE CANCEL DATA COMMAND: (2.17)

ASCII	DECIMAL	HEX
^X	24	18

THE LINE FEED COMMAND: (2.18)

ASCII	DECIMAL	HEX
^J	10	0A

THE FORM FEED COMMAND: (2.19)

ASCII	DECIMAL	HEX
^L	12	0C

6) GLOSSARY OF TERMS

This section is provided mainly for those unfamiliar with basic computer/printer terminology (i.e. byte, print head,...).

APPLESOFT: This is the version of BASIC that runs on the Apple II plus computer.

ASCII: This is an abbreviation for "American Standard Code for Information Interchange". This code is basically the definition of a character set in which numbers are assigned to characters (the lower and upper case alphabet, the digits 0-9, punctuation marks, mathematical symbols,...).

BIT: This is the smallest unit of information stored in a computer and has one of two values. These values are logic "0" and logic "1". An analogy to a bit is a digit in base ten, which holds one of ten values (0-9).

BYTE: This is a collection of eight bits. A byte is analogous to a number in base ten which is a collection of digits. Just as numbers hold values, so do bytes. Because character sets are defined in terms of assigning values to a set of characters, and bytes hold values (0-255), a byte can "hold a character".

DEFAULT: This simply means, "assumed if nothing else specified".

HEXADECIMAL: (hex). This is a numbering system based on 16 symbols (0-9, and A-F), just as decimal (base ten) is a numbering system based on ten symbols (0-9). In this numbering system bytes can be represented by two hexadecimal "digits".

PRINT HEAD: This is the part of the printer which moves back and forth horizontally and prints the characters on the paper.